CS112 - Fall 2022

Lab19

Instructor: Paul Haskell

INTRODUCTION

In this lab, you will work with some of the Random library components you learned about. We will all do some experiments with Bubble Sort, and we will write an automated test program.

Back Alley Dice

This is a simplified version of the Casino game craps. The game is played with two dice—we can use the Dice software we wrote a week ago. Back Alley Dice is played as follows:

- The player starts with a Stack of $20
- She removes $1 from her Stack to place her bet and then rolls two dice
- If the total score is 7 or 11, the player gets back 2x her original bet i.e. gains $1 overall
- If the total score is 2, 3, or 12, the player loses her original bet
- If the total is anything else, the total is called the "point". The player keeps on rolling until
    - She rolls a 7 before she rolls her point again, in which case she loses her original bet
    - She rolls her point again before a 7, in which case she wins 2x her original bet

Write a program called **BackAlley.java** that plays Back Alley Dice. Start with $20, bet $1 per game, and after every game print:

- Total money remaining and "win" or "lose"

For example

$21 win
$20 lose
$19 lose
$20 win
etc

Does your game ever play forever? What is the largest amount of money your Stack ever reached?

Bubble Sort

The **Lab19** directory in the CourseInfo repository has a Java Bubble Sort program for you, called **Bubble.java**. This program generates 10,000 random numbers, sorts them with a bubble sort, saves them to a file called "**sort.txt**", and prints to `System.err` the number of seconds required to perform the sort.

Your next task is to translate the Java Bubble Sort to Python, and write a program called **Bubble.py** . Use this program to generate 10,000 random numbers, sort them, and print out how much time was required for the sort. Do not use Python's built-in sort() methods--they use a smarter sorting algorithm than Bubble Sort, an algorithm we will discuss in 1-2 weeks.

Bubble.py should just print the number of seconds as a floating point number, with no units of time or text description, e.g.

20.593817

Create (probably by hand) a text file called **speed.txt** and in it write

Java,<<# of seconds required on your computer for Bubble.java to sort 10k elements>>

Python,<<# of seconds required on your computer for Bubble.py to sort 10k elements>>

I will collect and graph this data for all of us to review.

Test Software

Finally, please write an automated tester for **Bubble.java**, called **SortTest.java** . This program should read in the "**sort.txt**" file, ensure it contains the expected number of values, and ensure the values in the file are nondecreasing. If the file is good, print to `System.out`

PASS

If the file does not have the right number of elements, print

FAIL incorrect element count

If the file's elements are not sorted properly, print

FAIL incorrect sort

Reminder

Put all your files in Lab19 and push to GitHub before the deadline. This assignment must be turned in before 11:59pm Monday November 7

Conclusion

These programs are good preparation for our upcoming work on more advanced sorting techniques. Your **SortTest.java** will be useful to you then.

Grading Rubric

BackAlley.java is worth 20 points

- 10 points if it produces correct output
- 0 to 10 points based on the grader's evaluation of software and design quality.

Bubble.py is worth 10 points if it runs properly, prints out a time to sort, and if the time is reasonable.

Speed.txt is worth 5 points if it is present and contains the requested data.

SortTest.java is worth 12 points: 3 points for each of four test cases